

User Management / Directory Services using OpenLDAP

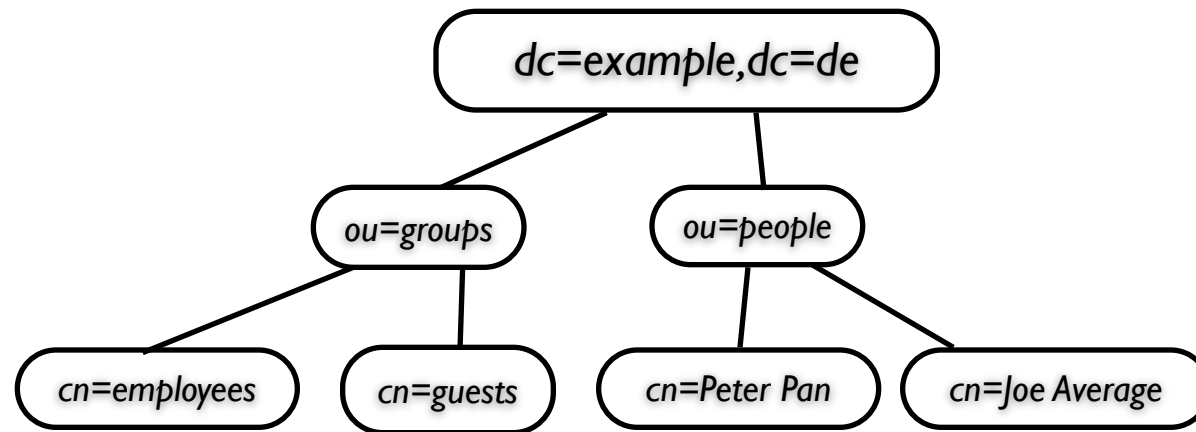
**Practical “The bioinformatics lab”
SS2010**

Benjamin Wellmann

- Introduction
 - LDAP-Structure
 - LDIF-Format
- Challenges
 - Setting up the server
 - Migrating User Management
 - Securing the connection using TLS

- Central User-database
- Central addressbook
- Organization of printers
- Administration of different units

Structure for Example AG



- DIT

- Entry: cn=Peter Pan,ou=people,dc=example,dc=de (dn)
 - cn - common name
 - ou - organizational unit
 - dc - domain component

dn: ou=People,dc=example,dc=de
ou: People
objectClass: organizationalUnit

Organizational
Unit
ou.ldif

dn: uid=averageJoe,ou=People,dc=example,dc=de
uid: averageJoe
cn: Average Joe
objectClass: account
objectClass: posixaccount
objectClass: top
userPassword:: e2NyeXB0fXg=
loginShell: /bin/bash
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/averagejoe

User Account
user1.ldif

Setting up the Server

```
aptitude install slapd ldap-utils
```

```
dpkg-reconfigure slapd
```

Omit OpenLDAP server configuration? -> No

DNS domain name: -> example.de

Organization name? -> example.de

Administrator password: -> password

Confirm password: -> password

Database backend to use: ->HDB (or BDB)

Do you want the database to be removed when slapd is purged? ->No

Allow LDAPv2 protocol? -> No

- Define tree structure DIT using LDIF-files
 - ou.ldif (people + group)
- Add users
 - user1.ldif (user-account)

```
sudo invoke-rc.d slapd stop #stops server sudo
```

```
slapadd -c -v -l /var/tmp/ou.ldif #adds ou.ldif
```

```
sudo invoke-rc.d slapd start #starts the server
```

```
ldapadd -c -x -D cn=admin,dc=example,dc=de -W -f  
/var/tmp/ou.ldif
```

- Check client configuration-file /etc/ldap/ldap.conf

```
BASE dc=example,dc=de #suffix / base of server
URI localhost #ip-adress of server
```

- Check the entries of the LDAP-database use ldapvi

```
ldapvi -d --host localhost
```

```
0 dc=example,dc=de
objectClass: top
objectClass: dcObject
objectClass: organization
o: noOrg
dc: example
```

```
1 cn=admin,dc=example,dc=de
objectClass:
simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

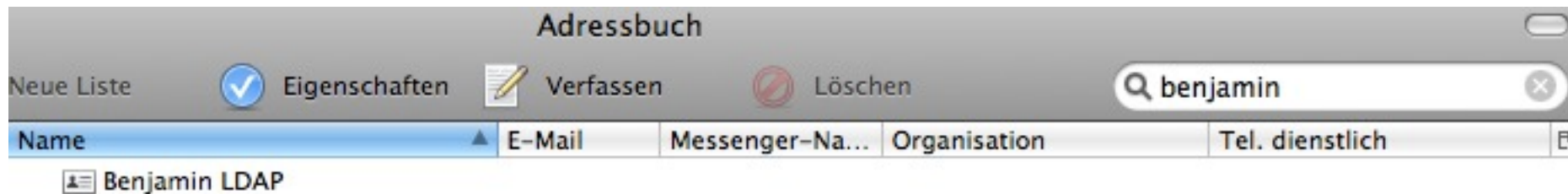

- Set up Thunderbird
 - IP-address, DN, port
- Start a search-query
- Read only?



The image shows the 'Allgemein' (General) tab of the Thunderbird LDAP configuration dialog. The fields are filled with the following values:

- Name: ldap_my
- Serveradresse: 172.16.101.156
- Basis-DN: dc=example,dc=de
- Port-Nummer: 389
- Bind-DN: (empty)

There is a 'Suchen' (Search) button next to the Basis-DN field. At the bottom, there is a checkbox for 'Verschlüsselte Verbindung (SSL) verwenden' (Use encrypted connection (SSL)) which is currently unchecked. The 'OK' button is highlighted in blue.



`libpam-ldap libnss-ldap migrationtools`

- libpam-ldap
 - PAM-modules for LDAP (API for authentication)
- libnss-ldap
 - Name Service Switch for LDAP
- migrationtools
 - migrate existing user-accounts to LDAP
- Create a NEW user on LDAP (not in local files)

- Check all files in /etc/pam.d/common-*
 - activated PAM-modules
 - should be already done by package-config
- Check all files in /etc/nsswitch.conf

```
#local user-management will be used first, second  
#the ldap-server
```

```
passwd: files ldap  
group: files ldap  
shadow: files ldap
```

- Connect using SSH or reboot (use NEW user-account)

- Transport Layer Security (similar to SSL)
 - encryption using public / private keys
- Create new certificates
 - `/usr/lib/ssl/misc/CA.sh - newca`

```
Country Name (2 letter code) [AU]:DE  
State or Province Name (full name) [Some-State]:Bavaria  
Locality Name (eg, city) []:Munich Organization  
Name (eg, company) [Internet Widgits Pty Ltd]: noOrg  
Organizational Unit Name (eg, section) []:Example  
Unit Common Name (eg, YOUR name) []:example.de #DN!!!  
Email Address []: peter@pan.edu
```

- We now have `cacert.pem` and `cakey.pem`

- Server certificate signing request (CSR)

```
openssl req -newkey rsa:1024 -nodes -keyout newreq.pem  
-out newreq.pem
```

- Let the CA cert sign the CSR

```
/usr/lib/ssl/misc/CA.sh -sign
```

- Move certs to desired path
- Create client certificate in a similar way
 - (different name)

- Edit slapd.conf

```
# CA signed certificate and server cert entries:
TLSCipherSuite TLS_RSA_AES_256_CBC_SHA #works best with debian

TLSCACertificateFile /var/openssl-data/cacert.pem
TLSCertificateFile /var/openssl-data/servercert.pem
TLSCertificateKeyFile /var/openssl-data/serverkey.pem

# Use the following if client authentication is required
TLSVerifyClient demand

# Or never desired
#TLSVerifyClient never
```

- Edit ldap.conf

```
# # Global LDAP settings #  
# See ldap.conf(5) for details  
# This file should be world readable but not world writable.
```

```
BASE dc=example,dc=de  
URI localhost
```

```
#location where your cacert is on the machine  
TLS_CACERT /home/user/certs/cacert.pem
```

```
TLS_REQCERT demand
```

- Test server

```
sudo invoke-rc.d slapd restart
```

```
openssl s_client -connect localhost:389 -showcerts -state -CAfile  
<ca cert>
```

Thanks for your attention...

Questions?